Co-processor Message Queues

How to talk to the VME system processor

Feb 7, 1989

The VME system software supports the connection of co-processors, which are additional cpu boards which are accessible from the VMEbus. Up to 15 co-processors may be supported. There is a system table which contains the parameters specific to each co-processor.

There is a simple message queue communications scheme that allows sending messages to a co-processor and accepting messages from one. A separate queue is used for each direction of communication. This note describes how a co-processor might find the message queues and use them.

The base location of the VME system table directory is at $00100000 on the VMEbus. The table directory contains an 8-byte entry for each system table numbered from 0–30. The Co-processor queue pointer table is table #15. As a result, the 8 bytes for this table at address $00100078 are found as follows:

word	(#entries)
word	(#bytes/entry)
longword	(Ptr to co-proc queue ptr table)

Using the co-processor#, perhaps obtained from some bits of the module status register set by switches on the 133A cpu board, find the corresponding entry in this table. Simply use as an offset the product of the co-proc# and the #bytes/entry word above. There will be found the following:

longword	(Ptr to co-processor command queue header)
word	(Size of queue)
word	(n.u.)
longword	(Ptr to co-processor readback queue header)
word	(Size of queue)
word	(n.u.)

The command queue is used to send messages *to* the co-processor, while the readback queue is used to receive messages *from* the co-processor.

At VME system reset time, the command queue is created by the VME system processor. Its 16-byte header's format is as follows:

word	IN	Offset to next message to go IN to queue
word	OUT	Offset to next message to be taken OUT of queue
word	LIMIT	Total size of queue including this header
word	START	Offset to first entry in queue (after header)
word	KEY	Key initialized to 'MZ' at reset time
byte	INErr	Diagnostic error count by IN user
byte	OUTErr	Diagnostic error count by OUT user
word	INCnt	Diagnostic count of messages placed INto queue
word	OUTCnt	Diagnostic count of messages taken OUT of queue

All offset words are offsets from the beginning of the header. At reset time, the KEY word is set to zero, while the queue is cleared and the header initialized. The IN, OUT, and START words are initialized to 16, the size of the header; thus, the queue contents immediately follow the header. The KEY word is set to 'MZ' as the last act of initialization.

The co-processor examines the command queue header when convenient. When it finds the value 'MZ' in the KEY word, it alters that word to 'MQ' to signal that it has recognized the command queue and will be monitoring it for messages. The VME system will not place any messages into the command queue unless it reads the value 'MQ' in the KEY word.

The diagnostic fields are there to allow inspection of queue activity by those interested.

When a message is placed into the queue, the IN offset is used to determine where to place it in the queue. There must be enough room so that the message can be stored as a single contiguous block in the queue, or it cannot be placed into the queue. The IN user must check the OUT and LIMIT offsets to be sure that there is available space for the message. If the IN offset is ≥ the OUT offset, and there is not enough room to place the message to end at least one word *before* LIMIT, then a word of zero is written at the IN offset, and the IN offset is reset to the START offset value. Then a further check must be made to see that the message will fit before the OUT offset.

After the message has been copied into place, the IN offset is advanced by the length of the message. Note that a message that causes the IN offset to be advanced to *equal* the OUT offset must not be entered into the queue, as the IN=OUT condition signifies an *empty* queue to the OUT user.

will always be even.

There are a few formats of messages that the VME system itself can generate. To allow for these cases, the second word of a message is the type word, which denotes the message type. The type word values used in such message will always be less than 256. Private communication messages can be sent to a co-processor by using type word values ≥ 256 and not conflict with those which the VME system may generate.

The following messages are sent by the VME system to a co-processor:

Analog Control

word	size =8
word	type =0–15, 128–143
word	index (any value)
word	data

This is used for making a "D/A setting" to a co-processor. The type value as well as the index value is stored in the analog control field of the analog descriptor for the channel. Bits 6–4 of the type word are used to specify the co-processor # used for determining which command queue is to receive the setting message. The remaining bits of the type word and the entire index word are arbitrary and have meaning only to the co-processor. They are designed to enable the co-processor to determine what device or pseudo-device it should set.

1553 Control

word	size (=10,12)
word	type (=15)
long	ptr to 1553 command block
word/long	setting data (2,4 bytes)

This is currently used for handling 1553 digital control. At a very low level, a decision is made *not* to handle the 1553 output *directly*, but instead to pass a message to a co-processor to handle it. The ramp co-processors in the Loma Linda accelerator system must have exclusive access to the 1553 controller hardware interface that controls the ramped power supplies. This scheme permits digital control to be handed over to the ramp cpu for execution when it has time during its 720 Hz activity.

In the future, one can expect that it will be possible to send a digital data word to a binary control word, denoted by type and index values analogous to those used for analog control.

word type ≥ 256
(any)

Listype #40 is used to send this form of message to a co-processor. The ident, in the long form, is as follows:

word lan-node
word co-processor# in range 0-15

It is the same as a channel ident, except that a co-processor number is used instead of a channel number.